



# **POLARIS U.K. LIMITED**

## **Opportunities offered by XML and Web Services**

**An information paper**

Author: Craig Sparkes & Ken Aldred

Version: 1.0

Status: Final

Date: 7 June 2002

## Amendment History

Version	Date	Author	Narrative
D1.4	23/5/2002	C Sparkes	Initial outline draft
D 1.5	31/5/2002	K Aldred	Revisions following discussions
F 1.0	7/6/2002	C Sparkes/K. Aldred	Distributed

## References

Doc Ref	Title	Version
IBM e-business June 2001	Dynamic e-business: Using Web Services to transform business - Mark Colan	

Distribution:

Polaris Board members

Polaris Board Advisory Group members

Polaris E-Market Review Group members

## Contents

<b>1</b>	<b>Introduction</b> .....	<b>4</b>
<b>2</b>	<b>Today</b> .....	<b>5</b>
<b>3</b>	<b>The Future</b> .....	<b>6</b>
3.1	Vision .....	6
3.2	Why is it within our grasp? .....	6
3.3	How might it affect our processes? .....	7
<b>4</b>	<b>Making it Possible</b> .....	<b>8</b>
4.1	A Common Understanding of Data .....	8
4.2	Definition of Web Services .....	8
4.3	Details for Developers .....	9
4.4	ProductWriter and RTE Software .....	10
<b>5</b>	<b>Polaris Actions</b> .....	<b>11</b>
5.1	The Master Components Library (MCL) .....	11
5.2	Transactional Messages .....	11
5.3	Dictionary Compatibility .....	11
<b>6</b>	<b>Conclusion</b> .....	<b>12</b>

# 1 Introduction

This paper is intended to provide a brief explanation of the new technologies that are now being implemented in all business sectors, including insurance. We are aware that some insurers are already taking an active interest in these technologies but the maximum benefits arise if there is an industry-wide approach to the core aspects. This paper summarises the technologies and the benefits they offer as well as how they can be used to make the insurance process more efficient.

Internet technologies, and in particular those based around XML give the insurance industry the opportunity to realise truly integrated processes. XML and the new XML based concept of 'Web Services' as well as giving us a way of tying together existing disparate computer applications, gives us a means to devise and build new distributed architectures - architectures in which the processing can be distributed across many applications and across many sites.

But what are Web Services? This is the term given to a number of new technologies which collectively make it possible to link applications in a standard and consistent manner. These technologies consist of a model for exchanging XML information, a language for describing the service (i.e. the function provided by an application), a language for describing the workflow between business partners or applications and a directory for finding new business partners.

This is a technical solution to a very real business problem, namely having the data and functionality needed for today's business needs locked into a variety of rigid bespoke systems designed for yesterday's requirements. Web Services provides a way to tie these information systems together both inside a company for system integration and externally for business partner integration.

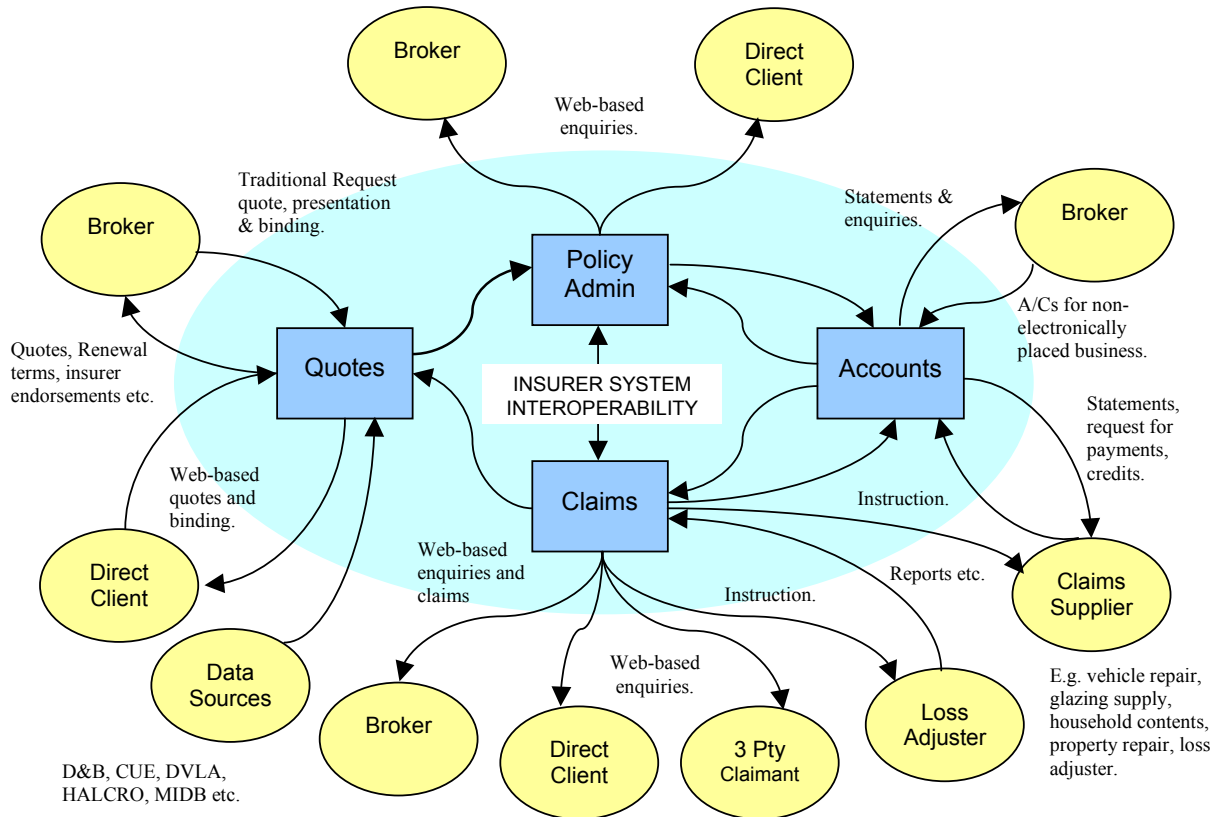
XML and Web Services are technologies that are being adopted on a global scale. All major software and IT service providers are committed to the development of the standards and are collaborating within the World Wide Web Consortium (W3C) to have them published as global recommendations. IBM, our new strategic partner for the infrastructure project, is putting significant effort into XML and Web Services – as is nearly every other major software vendor.

We should not under-estimate how far reaching the impact of this work could be. Web Services is not just about web site development, it is set to become the cornerstone of business to business trading and in-house integration.

## 2 Today

We know from discussions that the majority of interactions with external entities like the client, brokers, suppliers, loss adjusters, solicitors, government bodies, third party data providers etc are supported by systems and interfaces that have little automation or any form of consistency.

We suspect a high level model of all insurers' processes would look not unlike Figure 1. In fact it may be more complex than shown because, where we have drawn one box to represent a process, some companies may have two or more separate systems as a result of mergers and acquisitions etc.



**FIGURE 1:** Main Insurance Processes

Every arrow on the diagram represents a flow of information that is today exchanged using any number of methods – EDI, email, fax, paper forms, person to person contact etc. Most of these exchanges will require some form of manual interpretation and/or re-keying. Every step in the process causes delays, introduces errors and adds significant cost to the process chain. Removing or at least reducing this need to intervene at every point in the information chain is surely irresistible?

An analysis of these processes shows that the same or very similar tasks are repeated by many software systems and that many of the information exchanges between these systems are composed of essentially the same data. As well as the insurer systems shown, each of the parties: broker supplier, loss adjuster etc., have their own software applications that need to be considered.

## **3 The Future**

### **3.1 Vision**

The vision is to have all of these parties and systems linked together, working and cooperating in a seamless fashion; sharing data and processes where they are common, using specific or extended processes only where necessary or to provide competitive edge. Human intervention will be limited to what humans need to do or do best. Processing times, errors and omissions and the overall cost of doing business will all be reduced.

The physical location of where processes happen will become irrelevant because internet technologies will allow the integration of distributed software components, running on distant computer systems just as if they were all running on the same system. So, for example, the component parts of an automated process could reside where the maintenance expertise is, not where the users are.

Because processes and data will be shared and will be based upon industry standards, the cost of implementing new systems or integrating existing systems will be reduced - development / integration will be easier because system builders can base their work upon standard re-usable components and standard developer tools.

Most of this vision is not new, indeed EDI could be seen as the first step. What has changed is that the full vision is now within our grasp.

### **3.2 Why is it within our grasp?**

Two reasons. Firstly, major improvement in telecommunications standards such as the availability of cost effective broad-band facilities. Secondly, the computer industry has embraced open standards in the form of internet and XML protocols.

The first point is self-explanatory, but the second needs some elaboration.

There have been previous attempts to manage insurance information more efficiently. For example, EDI has been used for a number of years with some success – particularly for private motor transactions, but it has never really become a cornerstone technology. This is because EDI in its current form is really rather limited. It's an after the event way of sharing data, of avoiding re-keying and its associated errors.

It has never really progressed past this use because it is essentially an “asynchronous” protocol, an off-line way of doing things – the EDI message is sent in the expectation that the recipient will process whatever information it contains in their own good time.

XML and in particular Web Services allows programs to work “synchronously”. This means that one computer can initiate a process on a different computer in the knowledge that it will get a response in real-time. It waits for the response and then continues on its way. The initiating computer or “Service Consumer” doesn't have to know how the process or “Service” on the other computer worked or what computer language the programs were written in, it only has to understand the interface, and that interface is XML.

This is really simple, but really powerful. It is the combination of this functionality with industry standard XML messages that gives us the potential to build applications that can communicate with the widest possible audience with the minimal possible effort – all in real-time.

### 3.3 How might it affect our processes?

We have a significant benefit over most other industries because we work with information. Information can be stored on computers and moved around electronically. We don't have raw materials or stock that needs to be managed and we don't have complex machinery that needs constant maintenance. We have to communicate with our clients, intermediaries and suppliers, but on the whole this is still largely about exchanging information. In short with the exception of perhaps banking, our industry is about as perfectly placed to take advantage of these IT developments as you can get.

Virtually everything we do could benefit from these new XML protocols. For example, all of the arrows on Figure 1 represent places where we are starting to use XML messaging. Each of these "data flows" is not just one message but a family of messages.

Imagine how this could affect a process such as requesting an insurance quote:

A broker could key the basic risk information into their system. They press a button and this initiates a quoting service on the insurer's system. This service in turn, could initiate other services. It could do a quick credit check with a third party, or pull up previous claims/loss information from another. It could check the postcode of the insured for geographical hazards with another, check the insurers own database for multi-product discounts or cross marketing. All this could happen together and be presented back to the broker as a single quote within seconds.

If the broker accepts the quote, the insurer application could post the data to the broker's policy administration system - which of course is exposed as a Web Service. At the same time, the insurer could file the risk detail with the motor insurance database or register DTI statistics – all using standard protocols and standard messages – and all real-time.

There could be other services provided for the broker e.g. they could retrieve details about an ongoing claim for one of their clients – rather than phoning the insurer directly. They could retrieve statements or send in requests for payment all without leaving their own application. The integration options are almost endless.

As always there are some cautionary caveats to the rosy picture this paints. Insurance data is complex, reflecting that it concerns tangible objects and real activities and events. This inevitably results in a large amount of data that can be organised in many ways, however the vision depends on a standard approach.

Polaris are developing XML structures that could become the acknowledged standard for XML and web services throughout the market. This obviously implies that a commitment to these standards is necessary by all participants. We acknowledge that standards impose a certain degree of constraint upon the user but in most situations the benefits to all parties more than compensate for the need to conform.

## 4 Making it Possible

So XML and Web Services will drive all these things, but what does the industry actually need to make it happen? In short it needs to:

- adopt a common understanding of industry data and its scope – which means standards
- define Web Services in terms of the data standards
- document and publish details of these standard services to enable them to be built by individual insurers, and other service providers, and integrated by service consumers in a consistent manner.
- update existing industry standard components such as Polaris's ProductWriter and RTE software to participate in this approach.

Obviously individual insurers will have to adapt existing or develop new software to provide these services. The rest of this section describes each of these points in a little more detail.

### 4.1 A Common Understanding of Data

If we consider the data being exchanged between parties and between systems we find the same data appears all over the place.

Imagine you want a quote for your car insurance. You get a quote from your broker who's using the ProductWriter software. You are quoted a premium. That premium appears in a ProductWriter format and is stored on the broker's system in its own format. You accept that quote and a message is sent to the insurer's policy administration system where the premium is stored again. This system then sends a message to the insurer's accounting system, MI system etc., and is stored again.

It goes on. You have an accident. You report the claim and take your vehicle to an insurer appointed garage – who, perhaps has already received an instruction from the insurer containing relevant policy data like cover and excesses. Having completed the work, the garage sends a request for payment to the insurer. These amounts get loaded into the insurer's claims and accounts system etc.

Next year, when the insurer offers you renewal terms (it was only a small accident) the data that was sent in by the garage (which had become the costs on the claims record) gets loaded into whatever system is responsible for the renewal – eventually this information will end up back in the broker system and back in front of the ProductWriter software.

The same information is used throughout the scenario by many different parties and systems and is quite probably represented in many different formats. What is needed is a common way of representing these components and XML has the answer. XML Schema allows us to define reusable components. We can define what a monetary amount, vehicle details or an address looks like. What is more we can define them in a way that they can be shared by the whole industry and used over and over again by all of the different business processes.

### 4.2 Definition of Web Services

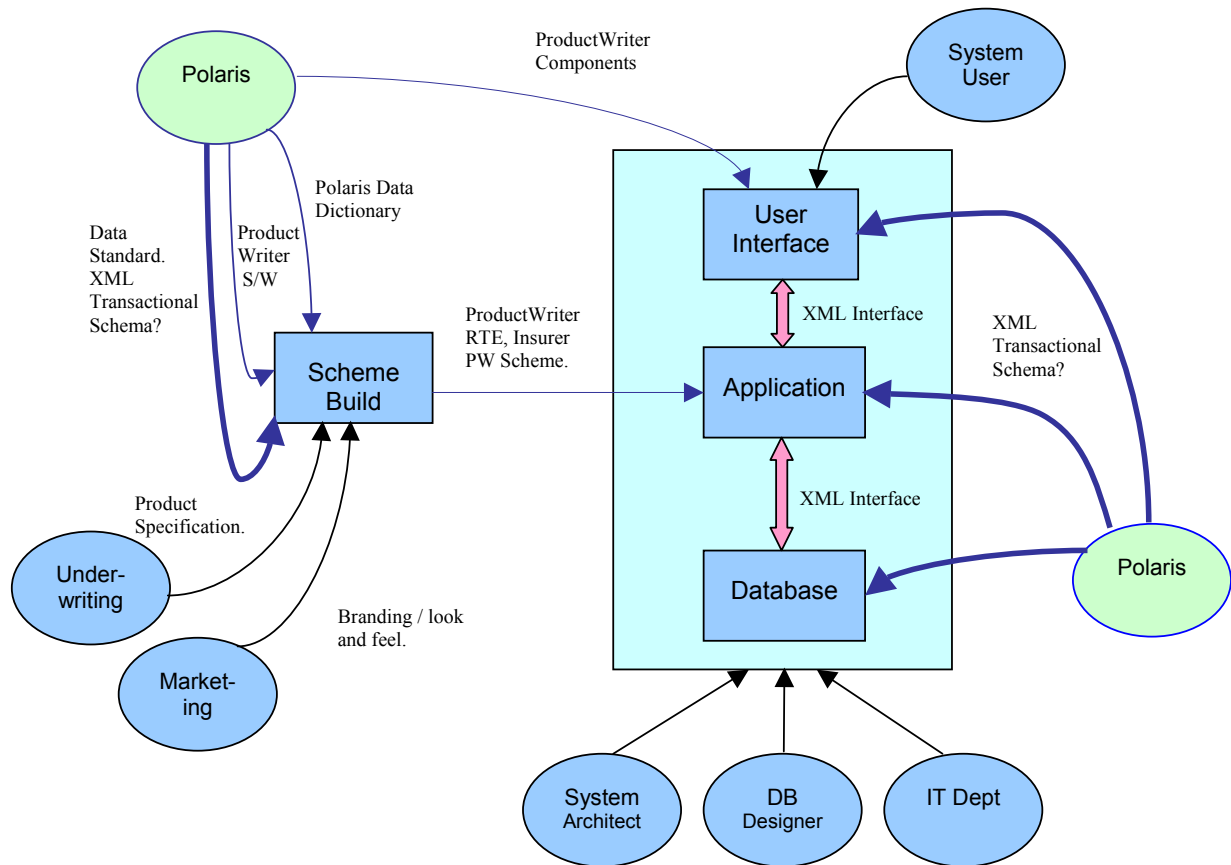
Remember, that a consumer of a web service doesn't have to understand how that service works, they only have to understand the interface – i.e. how to start the process, what information the process needs from them and what information the process will give back to them once it's done.

For example, the service might be the thing that actually does the motor quote in the previous example. In this case, it is going to need information about the drivers, vehicles and previous losses. When it's finished, it's going to tell you either that you are an unacceptable risk or its going to tell you how much the cover will cost and what terms and conditions are imposed.

This information is exchanged in an XML message. So to be sure that lots of users can be a consumer of the service, we need to describe the XML messages in a way that lots of people can understand; and we need to do so using the standard components mentioned about earlier. Again, this is the job of XML Schema.

### 4.3 Details for Developers

Whether you are building a web service application or a more traditional system, you need to do a similar set of activities and require a similar set of inputs. Figure 2 below shows a typical ProductWriter quotation system build.



**FIGURE 2:** Typical Quotation System Build

For quotes, insurers use a Polaris dictionary and associated data standards. In exactly the same way as a service consumer needs to know what data is input to and output from the service, an insurer building a product needs to know what data it will receive and what data it must output. In effect the insurer product could be thought of as the logic behind a web service.

Similarly the system builders need to know the same information, because they need to understand what data to pass to the insurer scheme and what data to store on the database.

Currently, this data definition is provided in various forms such as word processed documents and spreadsheets. XML Schema would be a more efficient way of doing this. The wide arrows on the diagram show where an XML Schema (identical to the web services ones) could be used and are described below:

- As input to the ProductWriter Toolset - to define the data that the ProductWriter user can use for a particular line of business.
- As input to the user interface build – either as a source of information or as electronic input into (say) an XFORMS development (see below).
- As a run-time validation component in the application processing – XML tools can validate that the data in an XML message matches the schema definition.
- As input to the database design process – the database must be capable of storing the data in XML messages received by the system.

Basically, the Schema are the key to linking all the various parties and functions in a consistent manner.

As the XML standards grow, we are seeing a widening of their functionality. XFORMS for example are W3C's approach to defining data capture characteristics and is the way that IBM are proposing to build service related forms on the E-Market infrastructure.

#### **4.4 ProductWriter and RTE Software**

At present, the XML-RTE allows an integrator to invoke a quotation etc., using an XML message and to receive a response in the format of an XML message, but that's about it. The XML messages to do this are tied tightly to the dictionaries and are not based upon standard components. The ProductWriter software itself, which is used to build and test insurer schemes is totally oblivious of XML.

To avoid ProductWriter and RTE software becoming remote from the vision of industry standard XML and Web Services it will need to become more supportive of XML. The RTE will need to be updated from its current bespoke XML to accept the new XML structures based on the industry standard components, and the ProductWriter toolset itself must become capable of accepting an XML Schema to provide transactional views of the dictionary.

The most effective way to make the sort of changes needed to align the Polaris software with this general direction will be the subject of debate over the coming months.

Updating our software in this way will bring most benefits if used with a single dictionary - in fact any approach to implementing a single dictionary cannot happen without the consideration of XML and *vice versa*. The options for developing a single dictionary are the subject of a separate document\*.

The very nature of a single dictionary implies that any one product will use only a subset of the overall dictionary. But how does a schemes builder know what parts of the dictionary to use? Clearly the logical approach would be to use an XML schema - the same schema as used to define our Web Service, user interface, databases etc.

\* Single Dictionary - Additional Feasibility Study (issued 24 May 2002)

## **5 Polaris Actions**

Polaris are laying the foundations for the industry to achieve this vision. We are in the process of defining the common components and standards that we need to have our software applications talk together and co-operate in a seamless fashion.

### **5.1 The Master Components Library (MCL)**

As a consequence of work on specific XML requirements (e.g. claims supplier messages) Polaris are developing a library or set of master components that will be used as the basis for future XML message development. These components will be devised using the XML schema grammar and will be available to all Polaris users for their own use.

At the same time, Polaris will stay in contact with other standards developers such as ACORD and ebXML with a view to sharing common components where practical.

### **5.2 Transactional Messages**

Polaris will develop business messages based upon these master components. Rules for deriving messages from the base components will be published separately.

### **5.3 Dictionary Compatibility**

A high degree of compatibility is required with Polaris dictionaries. Current expectations are that compatibility will be possible with the existing commercial dictionary but will probably not be possible with either the private motor or household data dictionaries due to their very specific structures. Polaris have put forward a proposal for a 'single dictionary' that will support all business classes including private motor and household and will correspond to the standards in the Master Component Library.

## **6 Conclusion**

We know from conversations with software builders from insurers, intermediaries and software houses that XML and Web Services are either being actively progressed or at least seriously considered. Although initially seen as a purely technical topic the implications are far reaching and this document has hopefully explained the potential benefits from a business perspective.

The maximum benefits can only be achieved by a consensus to develop and adopt the standards needed and to reinforce the overall direction towards data consistency and reuse. The opportunity for removing delays, improving quality and reducing costs is there to be taken.